

WE CLAIM:

1. A computer program product for controlling a source computer to update out-
of-date data stored by a plurality of destination computers with updated data stored by
said source computer using a computer network connecting said source computer to
said plurality of destination computers, said computer program product comprising:

associating code operable to associate priority data specifying a priority level
with each destination computer;

establishing code operable in dependence upon said priority data to establish a
plurality of groups of destination computers such that destination computers within a
group of destination computers share a common priority level;

generating code operable to generate a plurality of push update tasks driven by
said source computer, each push update task serving to transfer said updated data
from said source computer to a corresponding group of destination computers via said
computer network;

ordering code operable to order said plurality of push update tasks in
dependence upon said common priority level of each group of destination computers
to which a push update task relates to form a sequence of push update tasks such that
push update tasks corresponding to a higher common priority level occur within said
sequence before push update tasks corresponding to a lower common priority level;
and

execution code operable to sequentially execute said sequence of push update
tasks upon said source computer to transfer said updated data from said source
computer to said plurality of destination computers via said computer network.

2. A computer program product as claimed in claim 1, wherein said source
computer transfers said updated data to a group of destination computers using
multicast messages that are addressed to all destination computers within said group
of destination computers.

3. A computer program product as claimed in claim 2, wherein said computer
network uses an IP transmission protocol and said multicast messages are IP multicast
messages.

4. A computer program product as claimed in claim 1, wherein said updated data is one or more of:

malware definition data; and

a malware scanner program.

5. A computer program product as claimed in claim 4, wherein said malware includes one or more of computer viruses, worms, Trojans, banned files, banned words and banned images.

6. A computer program product as claimed in claim 1, wherein, if any group of destination computers includes more than a threshold number of destination computers sharing a common priority level, then said establishing code is operable to split said group to form a plurality of groups of destination computers from said destination computers sharing a common priority level and said ordering code is operable to order corresponding push update tasks to occur sequentially despite sharing said common priority level.

7. A computer program product as claimed in claim 6, wherein said splitting allocates destination computers sharing a common network portion of said computer network to a common group.

8. A computer program product as claimed in claim 6, wherein within said group of destination computers sharing a common priority level and being split, destination computers connected and not logged into said computer network are grouped together and split from and treated as having a lower priority level than destination computers connected and logged into said computer network.

9. A computer program product as claimed in claim 1, wherein if a push update task has not completed updating all destination computers within said corresponding group of destination computers within a timeout period, then said push update task is terminated and updating of destination computers not completed is added to a subsequent push update task.

10. A computer program product as claimed in claim 9, wherein a user alert message is generated identifying those destination computers for which updating was not completed.

11. A method of updating out-of-date data stored by a plurality of destination computers with updated data stored by a source computer using a computer network connecting said source computer to said plurality of destination computers, said method comprising the steps of:

associating priority data specifying a priority level with each destination computer;

in dependence upon said priority data, establishing a plurality of groups of destination computers such that destination computers within a group of destination computers share a common priority level;

generating a plurality of push update tasks driven by said source computer, each push update task serving to transfer said updated data from said source computer to a corresponding group of destination computers via said computer network;

ordering said plurality of push update tasks in dependence upon said common priority level of each group of destination computers to which a push update task relates to form a sequence of push update tasks such that push update tasks corresponding to a higher common priority level occur within said sequence before push update tasks corresponding to a lower common priority level; and

sequentially executing said sequence of push update tasks upon said source computer to transfer said updated data from said source computer to said plurality of destination computers via said computer network.

12. A method as claimed in claim 11, wherein said source computer transfers said updated data to a group of destination computers using multicast messages that are addressed to all destination computers within said group of destination computers.

13. A method as claimed in claim 12, wherein said computer network uses an IP transmission protocol and said multicast messages are IP multicast messages.

14. A method as claimed in claim 11, wherein said updated data is one or more of: malware definition data; and

a malware scanner program.

15. A method as claimed in claim 14, wherein said malware includes one or more of computer viruses, worms, Trojans, banned files, banned words and banned images.

5

16. A method as claimed in claim 11, wherein, if any group of destination computers includes more than a threshold number of destination computers sharing a common priority level, then said step of establishing splits said group to form a plurality of groups of destination computers from said destination computers sharing a common priority level and said step of ordering orders corresponding push update tasks to occur sequentially despite sharing said common priority level.

10

17. A method as claimed in claim 16, wherein said step of splitting allocates destination computers sharing a common network portion of said computer network to a common group.

15

18. A method as claimed in claim 16, wherein within said group of destination computers sharing a common priority level and being split, destination computers connected and not logged into said computer network are grouped together and split from and treated as having a lower priority level than destination computers connected and logged into said computer network.

20

19. A method as claimed in claim 11, wherein if a push update task has not completed updating all destination computers within said corresponding group of destination computers within a timeout period, then said push update task is terminated and updating of destination computers not completed is added to a subsequent push update task.

25

20. A method as claimed in claim 19, wherein a user alert message is generated identifying those destination computers for which updating was not completed.

30

21. Apparatus for updating out-of-date data stored by a plurality of destination computers with updated data stored by a source computer using a computer network

connecting said source computer to said plurality of destination computers, said apparatus comprising:

associating logic operable to associate priority data specifying a priority level with each destination computer;

5 establishing logic operable in dependence upon said priority data to establish a plurality of groups of destination computers such that destination computers within a group of destination computers share a common priority level;

generating logic operable to generate a plurality of push update tasks driven by said source computer, each push update task serving to transfer said updated data
10 from said source computer to a corresponding group of destination computers via said computer network;

ordering logic operable to order said plurality of push update tasks in dependence upon said common priority level of each group of destination computers to which a push update task relates to form a sequence of push update tasks such that
15 push update tasks corresponding to a higher common priority level occur within said sequence before push update tasks corresponding to a lower common priority level; and

execution logic operable to sequentially execute said sequence of push update tasks upon said source computer to transfer said updated data from said source
20 computer to said plurality of destination computers via said computer network.

22. Apparatus as claimed in claim 21, wherein said source computer transfers said updated data to a group of destination computers using multicast messages that are addressed to all destination computers within said group of destination computers.

23. Apparatus as claimed in claim 22, wherein said computer network uses an IP transmission protocol and said multicast messages are IP multicast messages.

24. Apparatus as claimed in claim 21, wherein said updated data is one or more of:
30 malware definition data; and
a malware scanner program.

25. Apparatus as claimed in claim 24, wherein said malware includes one or more of computer viruses, worms, Trojans, banned files, banned words and banned images.

26. Apparatus as claimed in claim 21, wherein, if any group of destination computers includes more than a threshold number of destination computers sharing a common priority level, then said establishing logic is operable to split said group to form a plurality of groups of destination computers from said destination computers sharing a common priority level and said ordering logic is operable to order corresponding push update tasks to occur sequentially despite sharing said common priority level.

27. Apparatus as claimed in claim 26, wherein said splitting allocates destination computers sharing a common network portion of said computer network to a common group.

28. Apparatus as claimed in claim 26, wherein within said group of destination computers sharing a common priority level and being split, destination computers connected and not logged into said computer network are grouped together and split from and treated as having a lower priority level than destination computers connected and logged into said computer network.

29. Apparatus as claimed in claim 21, wherein if a push update task has not completed updating all destination computers within said corresponding group of destination computers within a timeout period, then said push update task is terminated and updating of destination computers not completed is added to a subsequent push update task.

30. Apparatus as claimed in claim 29, wherein a user alert message is generated identifying those destination computers for which updating was not completed.